



USAISEC

AD-A268 237



2

*US Army Information Systems Engineering Command
Fort Huachuca, AZ 85613-5300*

U.S. ARMY INSTITUTE FOR RESEARCH
IN MANAGEMENT INFORMATION,
... COMMUNICATIONS, AND COMPUTER SCIENCES

DTIC
ELECTE
AUG 12 1993
S A D

The Software Support Qualitative Assessment Methodology

Volume V

Implementing the Operational Readiness Measure

This document has been approved
for public release and sale; its
distribution is unlimited.

March 1991

ASQB-GI-91-020

93-18866



**AIRMICS
115 O'Keefe Building
Georgia Institute of Technology
Atlanta, GA 30332-0800**



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704--188
Exp. Date: Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS NONE		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION / AVAILABILITY OF REPORT N/A		
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) ASQB-GI-91-020			5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A		
6a. NAME OF PERFORMING ORGANIZATION The Center for Information Management Research		6b. OFFICE SYMBOL (if applicable)		7a. NAME OF MONITORING ORGANIZATION N/A	
6c. ADDRESS (City, State, and ZIP Code) SERC Georgia Institute of Technology Atlanta, GA 30332		Dept of MIS University of Arizona Tucson, AZ 85613		7b. ADDRESS (City, State, and Zip Code) N/A	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION AIRMICS		8b. OFFICE SYMBOL (if applicable) ASQB - GI		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code) 115 O'Keefe Bldg., Georgia Institute of Technology Atlanta, GA 30332-0800		10. SOURCE OF FUNDING NUMBERS			
		PROGRAM ELEMENT NO. 62783A		PROJECT NO. DY10	TASK NO. 02-01-01
				WORK UNIT ACCESSION NO.	
11. TITLE (Include Security Classification) The Software Support Qualitative Assessment Methodology Volume V (UNCLASSIFIED) Implementing the Operational Readiness Measure					
12. PERSONAL AUTHOR(S) W. Michael McCracken, Elizabeth Mynatt, Christopher Smith (GIT) J.F. Nunamaker, Ai-Mei Chang, Titus Purdin, Richard Orwig, Amit Vyas (Univ of Arizona)					
13a. TYPE OF REPORT final report		13b. TIME COVERED FROM _____ TO _____		14. DATE OF REPORT (Year, Month, Day) 1991, March, 22	
				15. PAGE COUNT 50	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) life cycle metrics; systems support; support measures; qualitative assessment; supportability measures; assessment measures; readiness measures; information systems; software maintenance; support tools; tools management;		
FIELD	GROUP	SUB-GROUP			
19. ABSTRACT (Continue on reverse if necessary and identify by block number) The Software Supportability Qualitative Assessment Methodology is a five volume reference set that provides measures to aid in the support of information systems. The volumes are aimed at improving the support process by more accurately assessing the capabilities of support organizations, qualitatively measuring the supportability of fielded systems and evaluating the operational readiness of fielded systems. The five volumes are: I. Developing Quality Measures for Information Systems Support II. The Review of Metrics for Developing an Information Systems Support Measurement Framework III. Implementing the Software Supportability Measure IV. Implementing the Support Organization Assessment Measure V. Implementing the Operational Readiness Measure This volume provides guidelines for improving the operational readiness of an information system based on its evaluation. Specifics are provided for resource estimation for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring operational readiness.					
20. DISTRIBUTION / AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED / UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL Howard C "Butch" Higley			22b. TELEPHONE (Include Area Code) (404) 894-3110		22c. OFFICE SYMBOL ASQB-GI

The research herein was performed for the Army Institute for Research in Management Information, Communications, and Computer Sciences (AIRMICS), the RDTE organization of the U.S. Army Information Systems Engineering Command (USAISEC). The sponsor for the project was the Office of the Director of Information Systems for Command, Control, Communications, and Computers (ODISC4). The principal investigator was from The Center for Information Management Research (CIMR), W. Michael McCracken of the Georgia Institute of Technology.

This research report is not to be construed as an official Army position, unless so designated by other authorized documents. Material included herein is approved for public release, distribution unlimited, and is not protected by copyright laws. Your comments on all aspects of the document are solicited.

Accession For	
NTIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
A-1	

THIS REPORT HAS BEEN REVIEWED AND IS APPROVED

DTIC QUALITY INSPECTED 3

s/ _____
 Glenn E. Racine
 Chief
 CISC

s/ _____
 John R. Mitchell
 Director
 AIRMICS

The Software Support
Qualitative Assessment Methodology
Volume V
Implementing the Operational Readiness Measure

Prepared by
The Center for Information Management Research
for the
U.S. Army Institute for Research in Management
Information, Communications, and Computer Science
(AIRMICS)
Contract No. ECD-8904815

W. Michael McCracken, Elizabeth Mynatt, Christopher Smith
Software Engineering Research Center
Georgia Institute of Technology

December 1990

The **Software Supportability Qualitative Assessment Methodology** is a five volume reference set that provides measures to aid in the support of information systems. These manuals are aimed at improving the support process by more accurately assessing the capabilities of support organizations, quantitatively measuring the supportability of fielded systems and evaluating the operational readiness of fielded systems.

Volume I, *Developing Quality Measures for Information Systems Support*, describes the three measures along with the model of information system support that the measures are designed to satisfy. This is the main volume of the set and should be consulted before implementing the measures described in more detail in the other volumes.

Volume II, *The Review of Metrics for Developing an Information Systems Support Measurement Framework*, provides a survey and evaluation of current metrics in terms of information systems support. Specifically, three classes of metrics are reviewed: software product metrics, life cycle process metrics, and process management metrics.

Volume III, *Implementing the Software Supportability Measure*, provides instructions for collecting data for the measure, compiling the measure by evaluating the data, and interpreting the final result. The volume also contains guidelines for improving the supportability of an information system based on its evaluation. Specifically, the volume contains resource estimations for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring the supportability of an information system.

Volume IV, *Implementing the Support Organization Assessment Measure*, provides instructions for collecting data for the assessment, conducting the assessment, and interpreting the final result. The volume also contains guidelines for improving the capabilities of a support organization based on its evaluation. Specifically, the volume contains resource estimations for conducting and evaluating the assessment, questionnaires for collecting the required data and step-by-step instructions for measuring the capabilities of a support organization.

Volume V, *Implementing the Operational Readiness Measure*, provides instructions for collecting data for the measure, compiling the measure by evaluating the data, and interpreting the final result. The volume also contains guidelines for improving the operational readiness of an information system based on its evaluation. Specifically, the volume contains resource estimations for compiling and evaluating the measure, questionnaires for collecting the required data and step-by-step instructions for measuring the operational readiness of an information system.

Contents

1 Introduction	2
2 Requirements	2
Material	2
Audience	3
Time	3
3 Calculating and Evaluating the Operational Readiness Measure	4
4 Scoring the Questionnaire Answers	5
5 Interpreting the Results	5
The Operational Readiness Measure	6
The Supportability Factor	6
Reliability and Current State Factors	7
A Glossary of Terms	8
B List of Acronyms	11
C System Questionnaire	12
D Scoring Directions	13
E Operational Readiness Worksheet - Final Results	14

1 Introduction

The operational readiness measure for an information system is comprised of factors that affect an information system's operational readiness. Operational readiness is formally defined as follows:

Operational Readiness is the ability of a software system to perform its intended function upon request, based on:

- The current operational state of the system
- The reliability of the system
- The supportability of the system

Operational readiness addresses such questions such as "Will the system be up running when I need it?" and "When I use the system, can I expect correct results?" The information system support organization may utilize this measure not so much to determine the operational readiness of an information system at the present time as to forecast the operational readiness of an information system for the near future. Obviously, if the information system is presently in a state of disrepair, the operational readiness is "red" (or some other measure indicating a "down" state), and no additional measurements are necessary. However, there may be other situations in which we may wish to identify a critical operational readiness state. For example, a lack of adequate resources to support an information system combined with a large number of incoming maintenance requests for that system could indicate a critical state.

As indicated by the above definition, there are three factors of operational readiness. The *current state* factor measures components relating to current system operation and maintenance actions. The *reliability* factor measures components relating to the historical reliability of the information system. The *supportability* factor measures components related to the supportability of the information system. The supportability factor draws heavily upon the separately developed supportability measure.

This document describes how to measure the operational readiness of an information system that is maintained by your organization. The second section details what resources in time, material, and personnel are required to compute this measure. The following sections describe the process for computing and interpreting the operational readiness measure.

It is important you read through the following two sections (up through Calculating and Evaluating the Operational Readiness Measure) before beginning this process. It is also important that any personnel who will provide data for this process (by completing one or more questionnaires) do NOT read the sections on scoring the questionnaires and evaluating the results (Sections 4 and 5) until after they have completed the questionnaires.

2 Requirements

Material

Little in the way of materials is required to compile the operational readiness measure. Appendix C in this volume contains the system questionnaire, whose contents are the complete

list of questions to be answered for compiling the measure. All questions in the questionnaire must be answered. The questionnaire should be photocopied and distributed to the appropriate personnel (see next subsection). Appendix D contains directions and worksheets for scoring the questionnaires. A set of directions and a worksheet should be photocopied for each questionnaire. Appendix E contains a worksheet that should be used to record the final results of the operational readiness measurement. This worksheet should be photocopied and should be used in interpreting and evaluating the final measurement.

Audience

The careful selection of the appropriate personnel to complete the system questionnaires is critical to the success of the operational readiness measure. An examination of the questionnaire should be the first step in choosing your audience. In general, the questionnaire should be completed by personnel directly tasked with the actual maintenance of the information system.

Because a portion of the questionnaire requires subjective analysis or estimates (in cases where raw data is not available), distributing the questionnaires to more than one person involved with maintaining the information system and averaging the scores should reduce bias accompanying subjective responses.

Selecting a coordinator to distribute, collect, validate, and score the questionnaires is required. The coordinator is responsible for distributing the questionnaires and answering any remaining questions that the respondents may have. The coordinator must also collect the questionnaires, verifying that all questions have been answered completely. The coordinator must also validate the questionnaires against each other. Essentially the coordinator assures that the answers make sense (i.e. percentages add up to 100) and that the respondents interpreted the questions in the same manner. More information on this process can be found in the next section. Finally the coordinator is the best person to be responsible for scoring the questionnaire and compiling the final results. The coordinator may NOT complete any of the questionnaires as a respondent.

In summary, this process requires a minimum of two personnel to answer the system questionnaire and to serve as coordinator, respectively. The final results will be more meaningful if more than one person completes the system questionnaire.

Time

The amount of time required to compile the operational readiness measurement depends on two factors: the amount of on-line or easily accessible data and the number of personnel tasked to complete the system questionnaire. The questionnaire should take from 4 person-hours to 12 person-hours to complete. The amount of time required to actually complete the questionnaire depends on the availability and accessibility of system data. The questionnaire should take one half of a person-hour to score.

The amount of time required of the coordinator is determined by the number of personnel filling out the questionnaire. Validating the questionnaire should take approximately one person-hour per completed questionnaire. The effort should be less for a small number of questionnaires.

A rough formula to calculate the time required for collecting the data and calculating the measure is given below.

SW = Weight based on accessibility of information for the system.
Range from 1 (readily accessible information) to 3.

SN = Number of systems questionnaires completed.

TT = Total time required in person-hours.

$$TT = 4*SW*SN + (SN) + .5*(SN)$$

		- scoring
	- validation	
- completing system questionnaires		

3 Calculating and Evaluating the Operational Readiness Measure

This process consists of six steps.

1. Select respondents and coordinator.
2. Review questionnaire (optional).
3. Fill out questionnaire(s).
4. Validate questionnaire(s).
5. Score questionnaire(s), compute measure.
6. Interpret final result.

First, the personnel tasked with completing the questionnaires and the overall coordinator should be selected. Refer to the earlier section for guidelines for selecting questionnaire respondents. The system questionnaire included in this volume is the same questionnaire that is used to compile the supportability measure for an information system (see Volume III). Whereas for the supportability measure only certain questions in this questionnaire are completed, respondents must complete *all* questions in this questionnaire for the operational readiness measure.

Optionally, after the respondents have been selected, a meeting could be held to review the questionnaire. This meeting should be led by the coordinator. The purpose of this meeting is to assure that the respondents understand the questions in the same manner. Discussions about

possible answers should not be permitted. Only definitional information should be distributed. The advantage of this meeting is that it should quicken the completion of the questionnaires by the respondents and it should reduce the variability in their interpretation of the questions.

Next, the system questionnaire should be completed. The coordinator should be available to answer questions of interpretation. Respondents should be encouraged to write comments concerning interpretation next to their answers. This effort will aid the coordinator in validating the questionnaires.

The completed questionnaire(s) should be returned to the coordinator who should attempt to validate the responses. First, all questions should be answered! Second, responses containing quantitative, non-subjective data should correspond closely if not be equivalent. Third, the coordinator should look for differing interpretations by examining comments added by the respondents.

Next, the coordinator should score the questionnaire(s). Refer to the next section and appendices D and E for scoring directions. The system questionnaire scoring directions contains directions for scoring the questions, all of which must be answered to compute the operational readiness measure.

If more than one questionnaire is completed for an information system, the coordinator should average the scores to compute a final score. Finally, the coordinator and other personnel should consult Section 5 in order to interpret the final result.

4 Scoring the Questionnaire Answers

Appendices D and E contain directions for scoring the individual questionnaires and computing the final measure. Essentially, each answer will correspond to a certain number of points. Scoring directions for each question are provided with possible ranges specified. The results for each questionnaire are then recorded on the questionnaire worksheets. The worksheets divide the responses into three categories: Supportability, Reliability, and Current State. These categories are the three major factors of the operational readiness measure. The columns for each category should be totaled. If more than one questionnaire was completed for a given information system, the column totals for the respective questionnaires should then be averaged by the number of system questionnaires completed for the information system. These averages should be recorded on the Operational Readiness Worksheet. Instructions on the Operational Readiness Worksheet should then be followed to compute the final result.

5 Interpreting the Results

Operational readiness is a measure of the system to operate correctly on demand. The measure is predictive, indicating the ability of the information system to operate correctly on demand in the near future. Thus, operation readiness depends on the "current state" of the maintenance of the system as well as system reliability. In addition, operational readiness is heavily dependent upon the supportability of a system, since a system which is difficult to support is more likely to have operational difficulties when the need for expected or unexpected maintenance actions arises. Factors affecting the operational readiness of an information system are therefore divided

into three categories: supportability, reliability, and current state. The operational readiness measure is calculated such that supportability has about twice as much influence on the final value of the measure as the other two categories combined.

The Operational Readiness Measure

The operational readiness measure for an information system can be interpreted in more than one way. First, it can be interpreted as the percentage chance that a system will be "operationally ready" over the next period of time (a period of a month or less, but the period may vary depending upon the frequency of maintenance activities conducted on the system). This interpretation is rather fine-grained, however, and it may be difficult to interpret accurately.

An alternative interpretation would be to assign a label to operational readiness based upon the final score. One interpretation of operational readiness (an interpretation that has been used to assess the operational readiness of hardware equipment) is the use of the values "red," "yellow," and "green." A value of "red" indicates that the system is in a "down" state; that is, the system is currently not operational (at least as intended), or it will not be operational in the near future. A value of "yellow" indicates that, while the system is operational, there are significant problems that may threaten to bring the system down in the near future. A value of green indicates that the system is up and running as expected with no significant problems likely in the near future.

If we view operational readiness as a combination of the three factors of supportability, reliability, and current state, and we scale the total of all scores to fit within the range of 0-100, then the following interpretation of operational readiness may be useful:

71-100 Green

31-70 Yellow

0-30 Red

The Supportability Factor

An alternative approach of operational readiness is to examine the supportability factor and to examine the combination of the reliability and current state factors. (Note that because the value assigned to supportability is approximately equal to the combined value assigned to reliability and current state, we treat reliability and current state together). The final raw score for the supportability factor can range from 2 to 150. A useful interpretation for operational readiness from this perspective is as follows:

105-150 Green

45-104 Yellow

2-44 Red

The criteria that comprise the supportability factor of operational readiness include the size, age, complexity and modularity of the source code, modification history, existence and adequacy

of system documentation, existence of debugging code and adequate testing, length of support of the system, and the size and urgency of change requests.

Depending on the severity of the rating, the following actions may be taken to improve the supportability factor. This list does not represent all possible appropriate actions.

- Re-design the system (reverse engineering).
- Create or update the system documentation.
- Increase time spent on system testing.
- Accept fewer emergency change packages.
- Accept fewer large-scale changes.
- Consistently use regression testing.
- Increase the number and/or training of people supporting this system.
- Review the adequacy of software/hardware platform support.

Reliability and Current State Factors

Another view of operational readiness is through the combination of the reliability and current state factors. The combined raw score values for these two measures can range from 0 to 150. A useful interpretation from this perspective follows:

105-150 Green

45-104 Yellow

0-44 Red

The criteria comprising the current state and reliability factors include historical as well as current values of such items as number of maintenance requests already fulfilled or to be fulfilled, proportion of corrections and emergency requests, frequency of contact with users, number of staff currently supporting this system, and the estimated effort required to complete current changes to the information system.

The combined rating of reliability and current state can vary considerably from time to time, since maintenance requests are unlikely to originate at a uniform rate. Depending on the severity of the rating, the following actions may be taken to improve the current state and reliability factors. This list does not represent all possible appropriate actions.

- Increase number of staff to support this system, at least temporarily
- Review scheduling policy to determine if changes are needed.
- Redesign the system (for consistently low ratings).
- Accept fewer change requests.

A Glossary of Terms

Acceptance Review A review of a software product by developers and maintainers to determine if the product satisfies all originally specified requirements.

Acceptance Test Testing led by the client or QA group to determine whether the product satisfies its specifications as claimed by the developer.[Sch90]

Application System same as Information System

Availability A measure of the degree to which an item is in an operable and committable state at the start of a mission when the mission is called for at a random point in time.[Dep82]

Benchmark Testing Evaluation of the system performance against quantitative requirements.[Sch90]

Change Request Review Board An authority responsible for evaluating and approving requests for changes to a software product.

Cohesion A measure of the degree of the functional relatedness within program units. [Som89]

Complexity A characteristic of the software interface which influences the resources another system will expend or commit while interfacing with the software. [CDS86]

Configuration Management The process of identifying and defining the configuration items (hardware/software units) in a system, controlling the release and change of these items throughout the system life cycle, recording and reporting the status of configuration items and change requests, and verifying the completeness and correctness of configuration items.[IEE83]

Consistency The extent to which uniform design techniques and notation are used. [War87]

Coupling A measure of the strength of interconnections (dependencies) between program units. [Som89]

Error Human action that results in software containing a fault. Examples include omission or misinterpretation of user requirements in a software specification, incorrect translation or omission of a requirement in the design specification. [IEE83]

Failure A departure of program operation from program requirements.[IEE83]

Failure Rate The number of failures of an item per measure-of-life unit.[Dep82]

Fault A manifestation of an error in software. A fault, if encountered, may cause a failure. Synonymous with bug.

Fourth Generation Language (4GL) A computer programming language that provides abstractions of data and/or procedural specifications and is usually suited for a particular application domain.

Integration Testing Verify that the modules of the system combine correctly in order to achieve a product that meets its specifications. [Sch90]

IS (Information Systems) Organization An organized collection of procedures, personnel, and resources dedicated to support a portfolio of information systems.

Lines of Code Lines of source code, not including comments.

Maintainability The probability that an item will be retained in, or restored to, a specified condition within a given period if prescribed procedures and resources are used.[Dep82]

Maintenance All actions required to retain an item in, or restore it to, a specified condition.[Dep82]

Maintenance Audit An organized review of the maintenance organization.

Maintenance Escort Participation of the software maintainer in software system development.

Man/Machine Interface The software that supports the interaction between the user and the system.

Measure A high-level unit of specification which characterizes, evaluates, or predicts various aspects of software life cycle processes and products.

Metric A measurable indication of some aspect of a system. [DeM82] A quantification of a specific feature of the software life cycle process or software product.

Modularity A characteristic of software such that it is well-structured, highly cohesive, and minimally coupled. [War87]

New Systems Development The development of a system which has never been fielded.

Object Oriented Design Designing a system in terms of abstract data types where the objects are instantiations of the data types and new data types can be defined as extensions of previously defined types.

Regression Testing Testing the system against previous test cases to ensure that the functionality of the system has not been compromised by recent changes to the system. [Sch90]

Reliability The probability that an item will perform its intended function for a specified interval under stated conditions.[Dep82]

Self-Descriptiveness A characteristic of software that enables the understanding of implementation of software functions. [War87]

Support Staff The personnel tasked with maintaining an information system.

Supportability A measure of the adequacy of products, resources, and procedures to facilitate the support activities of modifying and installing software, establishing an operational software baseline, and meeting user requirements. [PTH87]

Testability The extent to which software facilitates both the establishment of test criteria and the evaluation of the software with respect to those criteria. [IEE83]

Throw-away prototyping Creating a prototype as part of system design and then "throwing away" the prototype and implementing the system "from scratch" not using any of the source code from the prototype.

Top-down design Designing the system by recursively breaking the system down into smaller components.

Unit Testing Testing of individual portions of the system.

B List of Acronyms

AIRMICS U.S. Army Institute for Research in Management Information, Communications,
and Computer Science

AMC Army Materiel Command

CCB Change Control Board

COE Army Corps of Engineers

FORSCOM Forces Command

HSC Army Health Services Command

IS Information System

ISC Army Information Systems Command

LOC Lines of Code

C System Questionnaire

This appendix contains a 7 page questionnaire for gathering information about the information system in order to calculate the Operational Readiness Measure. The questionnaire should be photocopied and distributed to selected respondents.

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire

QUESTIONNAIRE NUMBER: _____

Name of Information System: _____

Software and Documentation Information

1. What is the size of the system source code, in lines of code (LOC)?

_____ lines of code

2. What language(s) is the software written in?

3. How many modules (units that perform single functions or sets of functions) does the software product contain?

_____ number of modules

4. What is the age (measured from date of original installation) of the software product?

_____ age of system (in years)

5. How long has your organization supported this software product?

_____ length of support (in years)

6. What are the TOTAL number of changes that have been made to this product (software and associated documentation) during the time you have supported it? Include both Software Change Packages and Emergency Change Packages.

_____ total number of changes

7. Does the software contain any code that aids in debugging the software?

_____ yes
_____ no

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire

8. Is there any documentation explaining the overall function of the software?

_____ yes
_____ no

9. Is there documentation for each module explaining the module's function?

_____ yes
_____ no

10. Are there any user's manuals explaining the use of this software?

_____ yes
_____ no

Maintenance Information

11. For what amount of time (how many hours) during the month, if any, is the software system down and cannot be used?

_____ (hours) down time

12. What is the average number of maintenance requests per month received for this system?

[Notes: If a change proposal contains several requests, count each request separately.

Count ALL requests, even those that no actions are taken on.]

_____ average number of maintenance requests per month

13. Approximately how many of the above maintenance requests (per month) ultimately result in some change being made to the software?

_____ percentage of requests (per month) which result in changes to the software

14. Approximately what percentage of the maintenance requests FOR WHICH YOU PERFORM ACTIONS ON are

_____ Small-scale (affect a few lines of code at most)?

_____ Medium-scale (affect several functions or modules)?

_____ Large-scale (affect all or a large portion of the software)?

100 % TOTAL

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire

15. Approximately what percentage of the maintenance requests FOR WHICH
YOU PERFORM ACTIONS ON are

_____ EMERGENCY (require immediate attention and must be completed as
soon as possible to ensure the correct operation of the software)

_____ URGENT (require urgent attention -- more so than normal requests --
and must be completed within a relatively short period of time)

_____ NORMAL (require no special attention and can be completed within
the usual framework of support procedures)

100 % TOTAL

16. What percentage of ALL maintenance requests you receive...

_____ Are for corrections to faulty software components?

_____ Are for changes (other than corrections) or enhancements to the
software?

100 % TOTAL

17. What percentage (0-100%) of EMERGENCY and URGENT requests are for
corrections to faulty software components?

_____ percentage of EMERGENCY and URGENT requests that are corrections

18. ON THE AVERAGE, what percentage (0-100%) of all requests require more time
to complete than is originally scheduled?

_____ percentage of all requests completed behind schedule

19. What percentage of time spent maintaining the software is devoted to
testing it?

_____ (%) time spent on testing

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire

User Information

20. ON THE AVERAGE, how often do you communicate (either formally or informally) with a TYPICAL user organization using this information system? Mark the one appropriate response below.

_____ Several times a day
_____ Once or twice a day
_____ At least weekly, but not daily
_____ At least monthly, but not weekly
_____ At least once per year, but not monthly
_____ Less than once per year

Current Circumstances

21. How many people in your support organization presently maintain this software either on a part-time or full-time basis?
(Indicate the number in each category.)

_____ full-time (number)
_____ part-time (number)

22. AT PRESENT (NOT on the average), how many changes of all types (including corrections and enhancements) are there to be implemented?

_____ number of changes to be implemented

23. Of the above changes to be implemented, what percentage (0-100%) of these changes are EMERGENCY changes? If there are no changes, answer 0%.

_____ percentage of current changes that are EMERGENCY

24. Of the changes (from #2) to be implemented, what percentage (0-100%) of these changes are for CORRECTIONS to faulty software components? If there are no changes, answer 0%.

_____ percentage of current changes that are CORRECTIONS

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire

25. Based on the following scale, how you you rate the estimated effort needed to complete changes to the software product over the next month:

- 0 - Much more effort than average
- 1 - Somewhat more effort than average
- 2 - Average effort
- 3 - Less than average effort
- 4 - Much less than average effort
- 5 - No effort at all (no changes to implement)

_____ answer (0-5)

Problem Information

26. Overall, in your judgment, to what extent are (or have been) the following problems in maintaining this information system?
(Check the appropriate category.)

	No Problem At All	Somewhat Minor Problem	Minor Problem	Somewhat Major Problem	Major Problem
a. Not enough people to support this system.					
b. People supporting this system are not trained adequately.					
c. System is overly large, making support difficult.					
d. System is overly complex, making support difficult.					
e. System is not well-structured (written in "spaghetti code").					
f. Lack of system modularization makes changes difficult to implement.					

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire

26 (cont'd)

	No Problem At All	Somewhat Minor Problem	Minor Problem	Somewhat Major Problem	Major Problem
g. System is old and needs to be replaced.					
h. System documentation is incomplete or confusing.					
i. System documentation is out-of-date.					
j. Not enough time is spent on testing after changes are made.					
k. Software repair schedules are hard to meet.					
l. Overall, there are more change requests submitted for this system than can be handled.					
m. There are too many change requests resulting from software bugs (vs. enhancement requests).					
n. There are too many emergency change requests.					
o. User requirements for this system change frequently.					

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire

27. Overall, from your perspective, to what extent are (or have been) the problems as they impact on the ability to maintain this information system? (Check the appropriate category.)

	No Problem At All	Somewhat Minor Problem	Minor Problem	Somewhat Major Problem	Major Problem
a. Skills of maintenance programming personnel					
b. Number of maintenance programming personnel available					
c. Inadequate hardware/software configurations in IS Organization					
d. Motivation of maintenance programming personnel					
e. Maintenance programming productivity					
f. Competing demands between new systems development and maintenance					
g. Budgetary pressures					
h. Meeting scheduled commitments					

D Scoring Directions

This appendix contains the following two items:

1. A two-page worksheet for recording scores from the operational readiness questionnaire.
2. Twenty pages of directions for scoring the operational readiness questionnaire.

Software Supportability Qualitative Assessment Methodology Operational Readiness Worksheet

QUESTIONNAIRE NUMBER _____

Name of Information System: _____

Question	Supportability	Reliability	Current State	RANGE
1.	_____			0-5
2.	_____			0-5
3.	_____			0-5
4.	_____			0-5
5.	_____			0-5
6.	_____			0-5
7.	_____			0-2
8.	_____			0-2
9.	_____			0-2
10.	_____			0-2
11.		_____		0-10
12.			_____	0-5
13.			_____	0-10
14.	_____			1-4
15.	_____			1-4
16.		_____		0-20
17.		_____		0-5
18.*		_____	_____	0-5
19.	_____			0-4
20.	_____			0-10
21.			_____	0-5
22.			_____	0-20
23.		_____		0-10
24.*		_____	_____	0-10
25.			_____	0-10
TOTALS THIS PAGE	_____	_____	_____	

(Carry totals over to next page.)

* Enter the score in both of the indicated columns.

Software Supportability Qualitative Assessment Methodology Operational Readiness Worksheet

Question	Supportability	Reliability	Current State	RANGE
----------	----------------	-------------	---------------	-------

TOTALS FROM
LAST PAGE

26a.				0-5
b.				0-5
c.				0-5
d.				0-5
e.				0-5
f.				0-5
g.				0-5
h.				0-5
i.				0-5
j.				0-5
k.				0-5
l.				0-5
m.				0-5
n.				0-5
o.				0-5
27a.				0-5
b.				0-5
c.				0-5
d.				0-5
e.				0-5
f.				0-5
g.				0-5
h.				0-5

TOTALS

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

QUESTIONNAIRE NUMBER _____

Name of Information System: _____

1. What is the size of the system source code, in lines of code (LOC)?

_____ lines of code

SCORING

Calculate the score utilizing the following scale:

System Size			SCORE
At least	But less than		-----
0	10,000	lines of code	5
10,000	50,000	"	4
50,000	100,000	"	3
100,000	500,000	"	2
500,000	1,000,000	"	1
1,000,000		"	0

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

2. What language(s) is the software written in?

_____	_____
_____	_____
_____	_____

SCORING

Scoring should be based upon the following scale:

Number of languages	SCORE
-----	-----
1	4
2	3
3	2
4	1
Greater than 4	0

Add one additional point to the score if at least half of the languages are high-level, 4th generation languages or later.

Examples of allowable languages

Language	Allowed?
-----	-----
COBOL	No
Assembly	No
C	No
Ada	Yes
DBASE III	Yes
SQL	Yes

SCORE - _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

3. How many modules (units that perform single functions or sets of functions) does the software product contain?

_____ number of modules

SCORING

To calculate a score for this question, you need the answers for both this question and question number one (system size in lines of code).

Calculate the average module size, in lines of code, by dividing the answer in number one by the answer to this question. Then assign a score according to the following scale:

Average Module Size			
At Least	But Less Than		SCORE
-----	-----		-----
0	500	lines of code	5
500	1,000	"	4
1,000	2,000	"	3
2,000	3,000	"	2
3,000	5,000	"	1
5,000		"	0

Example: If the information system contains 200,000 lines of code and it contains 300 modules, then the average module size is:

$$200,000 / 300 = 1,250 \text{ lines of code.}$$

Thus, we would assign a score of 3.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

4. What is the age (measured from date of original installation)
of the software product?

_____ age of system (in years)

SCORING

Compute the score using the following scale:

System Age			
At Least	But Less Than		SCORE
-----	-----		-----
0	1	year	5
1	3	years	4
3	6	"	3
6	8	"	2
8	10	"	1
10		"	0

SCORE = _____

5. How long has your organization supported this software product?

_____ length of support (in years)

SCORING

Compute the score using the following scale (NOTE THAT THIS SCALE
IS DIFFERENT FROM THE ONE IN QUESTION 4):

Length of Support			
At Least	But Less Than		SCORE
-----	-----		-----
0	1	year	0
1	3	years	1
3	6	"	2
6	8	"	3
8	10	"	4
10		"	5

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

6. What are the TOTAL number of changes that have been made to this product (software and associated documentation) during the time you have supported it? Include both Software Change Packages and Emergency Change Packages.

_____ total number of changes

SCORING

To compute the score for this question, you need the answers for both this question and question number 5 (length of support).

Calculate the average number of changes per year by dividing the answer to this question by the answer in number 5. Then assign a score according to the following scale:

Average Number of Changes Per Year			SCORE
At Least	But Less Than		-----
0	5	changes per year	5
5	10	"	4
10	50	"	3
50	100	"	2
100	500	"	1
500		"	0

Example: If the information system has been supported for 5 years, and a total of 175 changes have been implemented to this system, then the average number of changes is:

$$175 / 5 = 35 \text{ changes per year.}$$

Thus, we would assign a score of 3.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

7. Does the software contain any code that aids in debugging the software?

_____ yes
_____ no

8. Is there any documentation explaining the overall function of the software?

_____ yes
_____ no

9. Is there documentation for each module explaining the module's function?

_____ yes
_____ no

10. Are there any user's manuals explaining the use of this software?

_____ yes
_____ no

SCORING

For each of questions 7 through 10, assign a score of 2 points for each "yes" answer and a score of 0 points for each "no" answer. For the four questions combined, a maximum score of 8 points is possible.

SCORE - _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

Maintenance Information

11. For what amount of time (how many hours) during the month, if any, is the software system down and cannot be used, on the average?

_____ (hours) down time

SCORING

Calculate the score utilizing the following scale:

Down time per month

At Least	But Less Than		SCORE
-----	-----		-----
0	1	hours	10
1	2	"	8
2	3	"	6
3	5	"	4
5	10	"	2
10		"	0

SCORE = _____

Software Supportability qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

12. What is the average number of maintenance requests per month received for this system?

[Notes: If a change proposal contains several requests, count each request separately.

Count ALL requests, even those that no actions are taken on.]

_____ average number of maintenance requests per month

SCORING

Calculate the score utilizing the following scale:

Avg. Number of Maintenance Requests			SCORE
At Least	But Less Than		-----
0	1	requests/month	5
1	5	"	4
5	10	"	3
10	50	"	2
50	100	"	1
100		"	0

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

13. Approximately how many of the above maintenance requests (per month) ultimately result in some change being made to the software?

_____ number of requests (per month) which result in changes to the software

SCORING

Calculate the score utilizing the following scale:

Avg. Number of
Maintenance Requests

At Least	But Less Than		SCORE
-----	-----		-----
0	1	requests/month	10
1	5	"	8
5	10	"	6
10	50	"	4
50	100	"	2
100		"	0

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

14. Approximately what percentage of the maintenance requests FOR WHICH
YOU PERFORM ACTIONS ON are

_____ Small-scale (affect a few lines of code at most)?
_____ Medium-scale (affect several functions or modules)?
_____ Large-scale (affect all or a large portion of the software)?

100 % TOTAL

SCORING

For each percentage, multiply by the corresponding weight
as shown in the following weight table:

Type of Action	WEIGHT
-----	-----
Small-Scale	4
Medium-Scale	3
Large-Scale	1

Sum the resulting products, divide the result by 100,
and round to the nearest integer. Maximum score is 4 points,
minimum score is 1 point.

Example: If the percentage for the various types of
maintenance requests are as follows:

45% Small-scale requests
45% Medium-scale requests
10% Large-scale requests

The calculation is:

$$\frac{(45 * 4) + (45 * 3) + (10 * 1)}{100} = 3.25,$$

100

which rounds to 3. Thus, the score is 3.

SCORE - _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

15. Approximately what percentage of the maintenance requests FOR WHICH
YOU PERFORM ACTIONS ON are

_____ EMERGENCY (require immediate attention and must be completed as
soon as possible to ensure the correct operation of the software)

_____ URGENT (require urgent attention -- more so than normal requests --
and must be completed within a relatively short period of time)

_____ NORMAL (require no special attention and can be completed within
the usual framework of support procedures)

100 % TOTAL

SCORING

For each percentage, multiply by the corresponding weight
as shown in the following weight table:

Type of Request	WEIGHT
-----	-----
Emergency	1
Urgent	2
Normal	4

Sum the resulting products, divide the result by 100,
and round to the nearest integer. Maximum score is 4 points,
minimum score is 1 point.

Example: If the percentage for the various types of
maintenance requests are as follows:

10% Emergency
10% Urgent
80% Normal

The calculation is:

$$\frac{(10 * 1) + (10 * 2) + (80 * 4)}{100} = 3.50,$$

which rounds to 4. Thus, the score is 4.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

16. What percentage of ALL maintenance requests you receive...

_____ Are for corrections to faulty software components?

_____ Are for changes (other than corrections) or enhancements to the software?

100 % TOTAL

SCORING

To calculate the score, divide the latter percentage figure (percentage of non-corrective changes) by 5 and round the result to the nearest integer. Maximum score 20 points, minimum score 0 points. (If the number of ALL maintenance requests you receive is 0, assume a score of 20).

Example: 30% corrections
70% enhancements

Calculation: $70 / 5 = 14$.

The score is 14.

SCORE = _____

17. What percentage (0-100%) of EMERGENCY and URGENT requests are for corrections to faulty software components?

_____ percentage of EMERGENCY and URGENT requests that are corrections

SCORING

Calculate the score as follows:

$(100 - \text{percentage}) / 20 = \text{Unrounded Score}$

Round the result to obtain a score between 0 and 5, inclusive.

Example: 65% of Emergency and Urgent requests are corrections.

Calculation: $(100 - 65) / 20 = 1.75$,

Which rounds to a score of 2.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

18. ON THE AVERAGE, what percentage (0-100%) of all requests require more time to complete than is originally scheduled?

_____ percentage of all requests completed behind schedule

SCORING

The calculation used to obtain this score is exactly the same as used in question 17. That is, apply

$$(100 - \text{percentage}) / 20$$

and round the result. (Score between 0 and 5, inclusive).

Example: 20% of requests are maintained behind schedule

Calculation: $(100 - 20) / 20 = 4.0$,

A score of 4.

SCORE = _____

19. What percentage of time spent maintaining the software is devoted to testing it?

_____ (%) time spent on testing

SCORING

To calculate the score for this question, utilize the following scale:

Percentage Test Time		SCORE
At Least	But less Than	
-----	-----	-----
0%	10%	0
10%	20%	1
20%	30%	2
30%	40%	3
40%		4

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

User Information

20. ON THE AVERAGE, how often do you communicate (either formally or informally) with a user organization using this information system? Mark the one appropriate response below.

	SCORE

_____ Several times a day	10
_____ Once or twice a day	8
_____ At least weekly, but not daily	6
_____ At least monthly, but not weekly	4
_____ At least once per year, but not monthly	2
_____ Less than once per year	0

SCORING

The list of scores corresponding to the selected item is shown above.

SCORE - _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

Current Circumstances

21. How many people in your support organization presently maintain this software either on a part-time or full-time basis?
(Indicate the number in each category.)

_____ full-time (number)
_____ part-time (number)

SCORING

Add the above two numbers together. Then, find the corresponding score in the following table:

TOTAL Number of Support Personnel	SCORE
-----	-----
0	0
1	1
2	2
3 - 5	3
6 - 10	4
Greater than 10	5

SCORE = _____

22. AT PRESENT (NOT on the average), how many changes of all types (including corrections and enhancements) are there to be implemented?

_____ number of changes to be implemented

SCORING

Calculate the score utilizing the following scale:

Number of Pending Changes		changes	SCORE
At Least	But Less Than		
-----	-----		-----
0	1	changes	20
1	5	"	16
5	10	"	12
10	50	"	8
50	100	"	4
100		"	0

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

23. Of the above changes to be implemented, what percentage (0-100%) of these changes are EMERGENCY changes? If there are no changes, answer 0%.

_____ percentage of current changes that are EMERGENCY

SCORING

To calculate the score, use the following formula:

(percentage) / 10

and round the result to obtain a score between 0 and 10, inclusive.

SCORE = _____

24. Of the changes (from #22) to be implemented, what percentage (0-100%) of these changes are for CORRECTIONS to faulty software components? If there are no changes, answer 0%.

_____ percentage of current changes that are CORRECTIONS

SCORING

To calculate the score, use the following formula:

(percentage) / 10

and round the result to obtain a score between 0 and 10, inclusive.

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

25. Based on the following scale, how you you rate the estimated effort needed to complete changes to the software product over the next month:

- 0 = Much more effort than average
- 1 = Somewhat more effort than average
- 2 = Average effort
- 3 = Less than average effort
- 4 = Much less than average effort
- 5 = No effort at all (no changes to implement)

_____ answer (0-5)

SCORING

To obtain the score, multiply the above answer by 2. Thus, if the answer above is 4, then the score is 8. (Maximum score 10 points, minimum score 0 points).

SCORE = _____

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

Problem Information

26. Overall, in your judgment, to what extent are (or have been) the following problems in maintaining this information system?
(Check the appropriate category.)

	No Problem At All	Somewhat Minor Problem	Minor Problem	Somewhat Major Problem	Major Problem	SCORE
a. Not enough people to support this system.						
b. People supporting this system are not trained adequately.						
c. System is overly large, making support difficult.						
d. System is overly complex, making support difficult.						
e. System is not well-structured (written in "spaghetti code").						
f. Lack of system modularization makes changes difficult to implement.						
g. System is old and needs to be replaced.						
h. System documentation is incomplete or confusing.						
i. System documentation is out-of-date.						
j. Not enough time is spent on testing after changes are made.						

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

26 (cont'd)

	No Problem At All					
Somewhat Minor Problem						
Minor Problem						
Somewhat Major Problem						
Major Problem						
k. Software repair schedules are hard to meet.						
l. Overall, there are more change requests submitted for this system than can be handled.						
m. There are too many change requests resulting from software bugs (vs. enhancement requests).						
n. There are too many emergency change requests.						
o. User requirements for this system change frequently.						

SCORING

For each lettered item, score

- 5 points for "No problem at all"
- 4 points for "Somewhat minor problem"
- 3 points for "Minor problem"
- 1 point for "Somewhat major problem"
- 0 points for "Major problem"

Software Supportability Qualitative Assessment Methodology
Operational Readiness Questionnaire - Scoring Directions

27. Overall, from your perspective, to what extent are (or have been) the problems as they impact on the ability to maintain this information system? (Check the appropriate category.)

	No Problem At All					
	Somewhat Minor Problem					
	Minor Problem					
	Somewhat Major Problem					
	Major Problem					
						SCORE

a. Skills of maintenance programming personnel						
b. Number of maintenance programming personnel available						
c. Inadequate hardware/software configurations in IS Organization						
d. Motivation of maintenance programming personnel						
e. Maintenance programming productivity						
f. Competing demands between new systems development and maintenance						
g. Budgetary pressures						
h. Meeting scheduled commitments						

SCORING

For each lettered item, score

- 5 points for "No problem at all"
- 4 points for "Somewhat minor problem"
- 3 points for "Minor problem"
- 1 point for "Somewhat major problem"
- 0 points for "Major problem"

E Operational Readiness Worksheet - Final Results

This appendix contains a one-page worksheet for calculating the final operational readiness measure.

Software Supportability Qualitative Assessment Methodology
Operational Readiness Worksheet

NAME OF SYSTEM _____

1. AVERAGE of the
Total Scores
 - a. Supportability _____
 - b. Reliability _____
 - c. Current State _____

2. AVERAGE Reliability plus
Current State
(1b + 1c) _____

3. AVERAGE Total Score
(1a + 1b + 1c) _____

4. DIVIDE Total Score above
by the number 3 for
Scaled Total _____

Example: AVERAGE Supportability Score: 120
AVERAGE Reliability Score: 60
AVERAGE Current State Score: 50

Reliability + Current State	110
Total Score	230
Scaled Total (rounded)	77

References

- [CDS86] S. D. Conte, H. E. Dunsmore, and V. Y. Shen. *Software Engineering Metrics and Models*. Benjamin/Cummings, Menlo Park, CA, 1986.
- [DeM82] T. DeMarco. *Controlling Software Projects*. Yourdon, New York, 1982.
- [Dep82] Department of Defense. *Test and Evaluation of System Reliability, Availability, and Maintainability: a Primer*, March 1982. DoD Directive 3235.1-H.
- [IEE83] IEEE. *IEEE Standard Glossary of Software Engineering Terminology*, 1983. Standard 729-1983.
- [PTH87] D. E. Peercy, E. Tomlin, and G. Horlbeck. Assessing software supportability risk: A minitutorial. In *Conference on Software Maintenance*, pages 72-80. IEEE, 1987.
- [Sch90] Stephen R. Schach. *Software Engineering*. Aksen Associates Incorporated Publishers, Boston, MA, 1990.
- [Som89] Ian Sommerville. *Software Engineering*. Addison-Wesley Publishing Company, New York, 1989.
- [War87] J. L. Warthman. Software quality measurement demonstration project (i). Technical Report RADC-TR-87-247, Rome Air Development Center, December 1987.